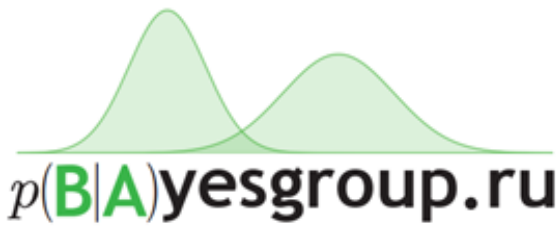
Bayes Group New Year Colloquium

Learning Structured Sparsity in Deep Neural Networks

Ashuha Arseniy[1,2]

Bayesian Research Group[1], MIPT[2]



ars-ashuha.ru/slides

December 27, 2016

Structured Sparsity Learning

- Structured Sparsity Learning

$$E(\mathbf{W}) = E_D(\mathbf{W}) + \lambda R(\mathbf{W}) + \lambda_g \cdot \sum_{l=1}^{L} \sum_{g=1}^{G} ||\mathbf{w}_g^l||_2$$

- Penalizing unimportant filers and channels

$$E(\boldsymbol{W}) = E_D(\boldsymbol{W}) + \lambda_n \cdot \sum_{l=1}^{L} \left(\sum_{n_l=1}^{N_l} ||\boldsymbol{W}^{(l)}_{n_l,:,:,:}||_g \right) + \lambda_c \cdot \sum_{l=1}^{L} \left(\sum_{c_l=1}^{C_l} ||\boldsymbol{W}^{(l)}_{:,c_l,:,:}||_g \right).$$

  N_l num filters, C_l num channels in l-th layer

- Learning arbitrary shapes of filers

$$E(\boldsymbol{W}) = E_D(\boldsymbol{W}) + \lambda_s \cdot \sum_{l=1}^{L} \left(\sum_{c_l=1}^{C_l} \sum_{m_l=1}^{M_l} \sum_{k_l=1}^{K_l} ||\boldsymbol{W}^{(l)}_{:,c_l,m_l,k_l}||_g \right).$$

- Regularizing layer depth

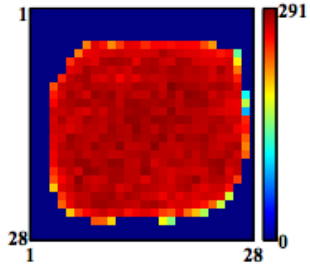
$$E(\boldsymbol{W}) = E_D(\boldsymbol{W}) + \lambda_d \cdot \sum_{l=1}^{L} ||\boldsymbol{W}^{(l)}||_g.$$

Experiments: LeNet and MLP on MNIST

- MLP sparse about 40%

MLP #	Error	Neuron # per layer [§]	FLOP per layer [§]
1 (*baseline*)	1.43%	784–500–300–10	100%–100%–100%
2	1.34%	469–294–166–10	35.18%–32.54%–55.33%
3	1.53%	434–174–78–10	19.26%–9.05%–26.00%

[§]In the order of *input layer–hidden layer 1–hidden layer 2–output layer*



- LeNet

Table 1: Results after penalizing unimportant filters and channels in *LeNet*

LeNet #	Error	Filter # [§]	Channel # [§]	FLOP [§]	Speedup [§]
1 (*baseline*)	0.9%	20—50	1—20	100%—100%	1.00×—1.00×
2	0.8%	5—19	1—4	25%—7.6%	1.64×—5.23×
3	1.0%	3—12	1—3	15%—3.6%	1.99×—7.44×

[§]In the order of *conv1—conv2*

Table 2: Results after learning filter shapes in *LeNet*

LeNet #	Error	Filter size [§]	Channel #	FLOP	Speedup
1 (*baseline*)	0.9%	25—500	1—20	100%—100%	1.00×—1.00×
4	0.8%	21—41	1—2	8.4%—8.2%	2.33×—6.93×
5	1.0%	7—14	1—1	1.4%—2.8%	5.19×—10.82×

[§] The sizes of filters after removing zero shape fibers, in the order of *conv1—conv2*

Experiments: ConvNet and ResNet on CIFAR-10

- ConvNet CIFAR10

Table 3: Learning row-wise and column-wise sparsity of *ConvNet* on CIFAR-10

ConvNet #	Error	Row sparsity §	Column sparsity §	Speedup §
1 (*baseline*)	17.9%	12.5%–0%–0%	0%–0%–0%	1.00×–1.00×–1.00×
2	17.9%	50.0%–28.1%–1.6%	0%–59.3%–35.1%	1.43×–3.05×–1.57×
3	16.9%	31.3%–0%–1.6%	0%–42.8%–9.8%	1.25×–2.01×–1.18×

§in the order of *conv1–conv2–conv3*

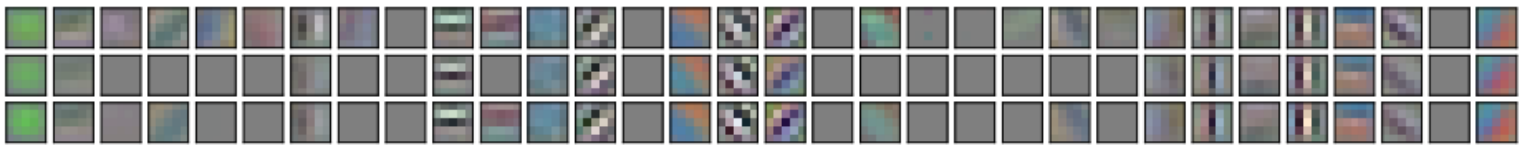


Figure 5: Learned *conv1* filters in *ConvNet 1* (top), *ConvNet 2* (middle) and *ConvNet 3* (bottom)

- ResNet sparse about 40%

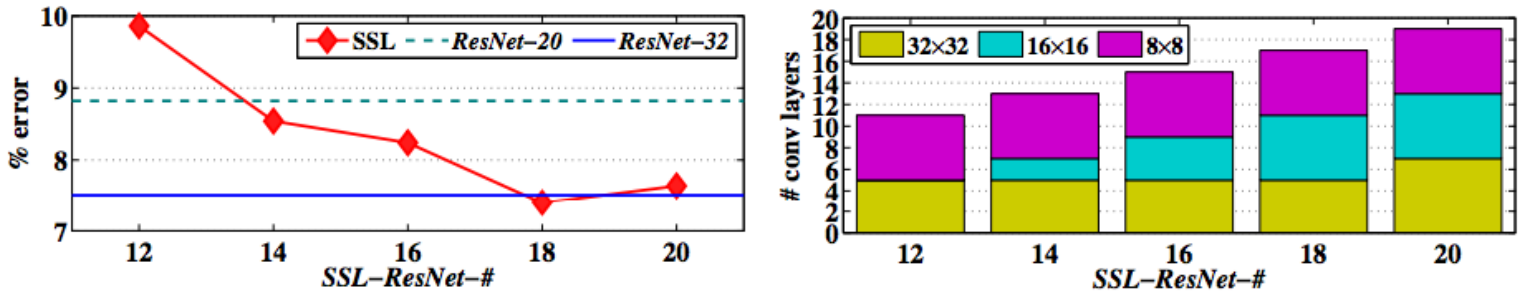
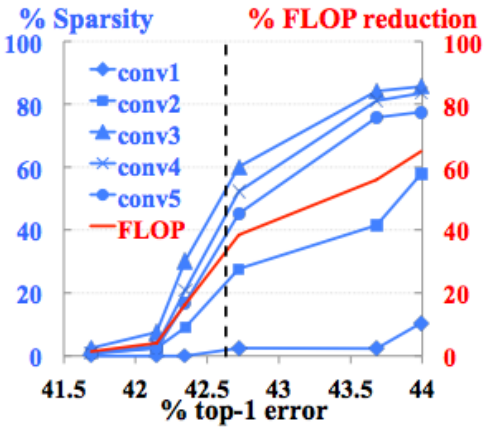
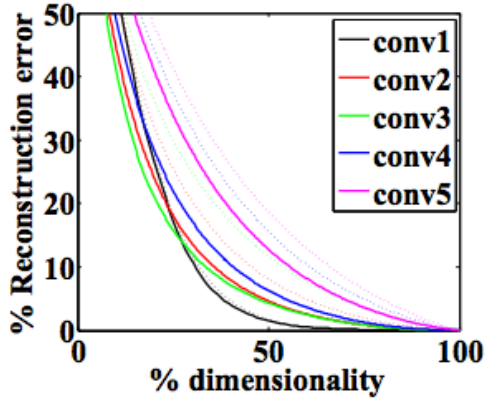
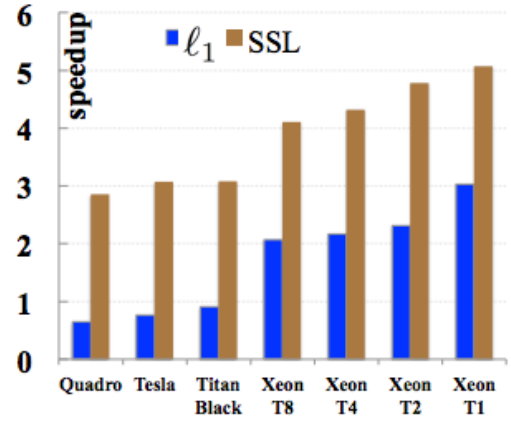


Figure 6: Error vs. layer number after depth regularization. # is the number of layers including the last fully-connected layer. *ResNet-#* is the *ResNet* in [5]. *SSL-ResNet-#* is the depth-regularized *ResNet* by SSL. 32×32 indicates the convolutional layers with an output map size of 32×32, *etc.*

Experiments: AlexNet on ImageNet







Bibliography

- Learning Structured Sparsity in Deep Neural Networks, NIPS 16
- Tree-guided group lasso for multi-task regression with structured sparsity, ICML10